

INTEGRATED CIRCUIT FAULT INSERTION SYSTEM

FIELD OF THE INVENTION

[0001] The present invention relates to fault insertion in an integrated circuit, and more particularly to fault
5 insertion for the purpose of testing diagnostic software.

BACKGROUND OF THE INVENTION

[0002] The traditional goal of circuit level testing is to determine that all components on a printed circuit board assembly (PCBA) are functioning properly. This
10 testing is typically performed at the manufacturing level so as to detect defects at the earliest stage, and is performed while the PCBA is in a non-functional state. The testing is accomplished by applying a set of test input signals at various nodes on the PCBA and reading a
15 resulting set of test output signals. It can be determined if the PCBA components are functioning properly by comparing the actual test output signals with the expected set of test output signals.

[0003] For PCBAs having a low level of integration, most nodes of the PCBA are externally accessible, and
20 testing can be accomplished by, for example, a "bed of nails" test jig. As PCBAs become increasingly more complex in their use of integrated circuits and surface mount interconnection technology, more of the nodes and
25 circuits of the PCBA become externally inaccessible, or hidden. In highly integrated circuits, such as field programmable gate arrays (FPGAs), programmable logic devices (PLDs), large scale, very large scale and ultra large scale integration (LSI/VLSI/ULSI), and application
30 specific integrated circuits (ASICs), different designs for testability and testing methodologies are required.

[0004] A well known and common manufacturing design technique allowing for easier testing of highly integrated circuits is the "boundary-scan" technique, as described in

the Institute of Electrical and Electronic Engineers (IEEE) standards document IEEE Std 1149.1-1990, "IEEE Standard Test Access Port and Boundary Scan Architecture," IEEE, 1993. With the boundary-scan technique, all

5 integrated circuit input and output pins comprise a scan cell in the signal path having a storage device and a switch. In normal operation, the switch connects the integrated circuit pins to the normal functional signals. In a test configuration, certain of the switches connect

10 pins to the storage devices, allowing stored values to be applied to input pins, and the reading of output signals from storage devices connected to output pins. All scan cells are connected to a serial bus that provides for control of all input and output operations of the

15 integrated circuit.

[0005] Boundary-scan design can be very effective for hardware testing, but is less so when applied to the testing of a system's diagnostic software. In high availability systems, diagnostic software is a critical

20 system component used to identify problems and, in some cases, to attempt corrective actions. It is very important to be able to test and verify a system's diagnostic software. A common method of testing diagnostic software is to intentionally induce faults in

25 the hardware while the system is in normal operating mode, see that the faults are properly identified by the diagnostic software and, where applicable, appropriate actions are taken. In many systems, it is possible to induce hardware faults through a system backplane by, for

30 example, grounding backplane pins. In highly integrated systems, however, many nodes and circuits are hidden. Another problem with manually inducing faults through the grounding of accessible circuit nodes is that certain interface logic families, for example, the Advanced BiCMOS

35 Technology (ABT) family of devices, have high drive

capabilities that can overpower a manually induced short to ground.

SUMMARY OF THE INVENTION

[0006] Accordingly, it is an object of the present invention to provide a system to induce faults in highly integrated circuits. A further object of the present invention is to provide a system that is independent of the boundary scan technique. A further object of the present invention is to provide a system that does not require faults be induced through manually grounding or forcing high accessible circuit nodes.

[0007] The present invention is a system for fault insertion in an integrated circuit that resides in a functional portion of the integrated circuit. The fault insertion system is microprocessor controlled through a Fault Control Register (FCR). The FCR comprises two registers: a Fault Identification Register (FIR), and a Fault Apply Register (FAR). The FIR is connected to a FIR decode block which, depending on the values contained in the FIR, applies signals to one or more node fault logic blocks connected to nodes of the integrated circuit. The node fault logic blocks either apply a test signal to a circuit node, or apply the normal system signals to the node. The FAR controls an enable signal to the FIR decode block, and determines when, and the duration, that the test signal will be applied. An External Control Bit of the FAR also allows manual control of the test signal.

DESCRIPTION OF THE DRAWINGS

[0008] FIG. 1 shows a block diagram of a preferred embodiment of the present invention.

[0009] FIG. 2 shows a block diagram of an embodiment of a force fail low node fault logic block.

[0010] FIG. 3 shows a block diagram of an embodiment of a force fail high node fault logic block.

[0011] FIG. 4A shows a block diagram of a first embodiment of a force fail high/low node fault logic block.

[0012] FIG. 4B shows a block diagram of a second embodiment of a force fail high/low node fault logic block.

DESCRIPTION OF THE PREFERRED EMBODIMENT

[0013] FIG. 1 shows a block diagram of a preferred embodiment of the present invention. All residing in the functional portion of an integrated circuit, Fault Control Register (FCR) 2 is comprised of a Fault Identification Register (FIR), and a Fault Apply Register (FAR) that includes an External Control Bit (EXT). The FIR is connected to FIR decode block 4. FIR decode block 4 has output signals FAULT_1 to FAULT_N, which are connected to integrated circuit node fault logic blocks. By way of illustration, one such node fault logic block is shown, and comprises signal FAULT_1 connected to one input of AND gate G3, normal system signal SIGNAL IN connected to the other input of AND gate G3, and AND gate G3 output signal SIGNAL OUT. Each of signals FAULT_1 through FAULT_N is connected to a node fault logic block, and each node fault logic block comprises one or more of signals FAULT_1 through FAULT_N. The FAR is connected to FAR decode block 3, which has an output connected to one input of AND gate G2. Output signal ENABLE of AND gate G2 is connected to FIR decode block 4. Bit EXT is connected to an inverted input of OR gate G1. The output of OR gate G1 is connected to the other input of AND gate G2.

[0014] External microprocessor 1 is connected to the FIR and FAR of FCR 2. An external manual switch SW1 is connected to the other inverted input of OR gate G1. Switch SW1 is normally in a first position and applies Vcc, or can be momentarily switched to a second position to apply a ground signal.

[0015] In operation of the embodiment illustrated in FIG. 1, microprocessor 1 loads one of a set of predetermined Fault Identification Code (FIC) values into the FIR of FCR 2. FIR decode block 4 decodes the FIC, which will result in signal FAULT_1 being asserted low while signal ENABLE is asserted to FIR decode block 4. In general, each of the set of predetermined FIC values results in FIR decode block 4 asserting an output signal comprising a combination of signals FAULT_1 to FAULT_N while signal ENABLE is asserted.

[0016] Signal ENABLE is asserted in one of two ways. In the automatic method of asserting signal ENABLE, switch SW1 is in its normal first position connected to Vcc, and bit EXT of the FAR is not set by microprocessor 1, resulting in OR gate G1 asserting its output to one of the inputs of AND gate G2. Microprocessor 1 then loads the predetermined fault-on Fault Apply Code (FAC) value into the FAR of FCR 2. FAR decode block 3 decodes the FAC, which results in the FAR decode block 3 output signal being asserted on the other input of AND gate G2, which further results in AND gate G2 output signal ENABLE being asserted to FIR decode block 4, which further results in signal FAULT_1 being asserted low. As illustrated in FIG. 1, the node fault logic block that controls the node connected to AND gate G3 is an example of a force fail low node fault block. When signal FAULT_1 is asserted low, signal SIGNAL OUT remains low regardless of the value of normal system signal SIGNAL IN. When microprocessor 1 loads the predetermined fault-off FAC value into the FAR of FCR 2, FAR decode block 3 decodes the FAC, which results in the FAR decode block 3 output signal being deasserted on the other input of AND gate G2, which further results in AND gate G2 output signal ENABLE being deasserted to FIR decode block 4, which further results in signal FAULT_1 being deasserted high. With signal FAULT_1

being deasserted high, SIGNAL OUT will have the same value as normal system signal SIGNAL IN.

[0017] In the manual method of asserting signal ENABLE, bit EXT of the FAR is set by microprocessor 1, resulting in OR gate G1 deasserting its output to one of the inputs of AND gate G2. Microprocessor 1 then loads the predetermined fault-on Fault Apply Code (FAC) value into the FAR of FCR 2. FAR decode block 3 decodes the FAC, which results in the FAR decode block 3 output signal being asserted on the other input of AND gate G2. When, for example, a tester momentarily moves switch SW1 to its second ground position, the output signal of OR gate G1 is asserted at AND gate G2, which further results in AND gate G2 output signal ENABLE being asserted to FIR decode block 4, which further results in signal FAULT_1 being asserted low. When switch SW1 is moved to its normal first Vcc position, the output signal of OR gate G1 is deasserted at AND gate G2, which results in AND gate G2 output signal ENABLE being deasserted to FIR decode block 4, which further results in signal FAULT_1 being deasserted high.

[0018] FIGS. 2, 3, 4A and 4B show examples of node fault logic blocks. FIG. 2 shows an example of a force fail low, or stuck-at-zero, node fault logic block. When FIR decode block 4 output signal FAULT_1 is deasserted high, AND gate G20 output signal SIGNAL OUT follows normal system signal SIGNAL IN. When signal FAULT_1 is asserted low, signal SIGNAL OUT remains low regardless of the value of system signal SIGNAL IN. FIG. 3 shows an example of a force fail high, or stuck-at-one, node fault logic block. When signal FAULT_1 is deasserted low, OR gate G30 output signal SIGNAL OUT follows normal system signal SIGNAL IN. When signal FAULT_1 is asserted high, signal SIGNAL OUT remains high regardless of the value of system signal SIGNAL IN. FIG. 4A shows a first example of a force fail high or low node fault logic block. In this circuit, fault insertion is controlled by two output signals from

5 FIR decode block 4. When signal FAULT_1 is deasserted low and signal FAULT_2 is deasserted high, signal SIGNAL OUT follows system signal SIGNAL IN. When signal FAULT_1 is asserted high and signal FAULT_2 is deasserted high, signal SIGNAL OUT remains high regardless of the value of signal SIGNAL IN. When signal FAULT_2 is asserted low, signal SIGNAL OUT remains low regardless of the value of signal SIGNAL IN. FIG. 4B shows a second example of a force fail high or low node fault logic block. When signal FAULT_1 is deasserted high and signal FAULT_2 is deasserted low, signal SIGNAL OUT follows system signal SIGNAL IN. When signal FAULT_1 is asserted low, signal SIGNAL OUT follows signal FAULT_2.

15 **[0019]** In the present invention, microprocessor 1 can be programmed to induce permanent faults, transient faults and intermittent faults through fault assertion codes applied to the FAR of FCR 2. As disclosed above, a tester may also control fault assertion through switch SW1.

20 **[0020]** In the node fault logic block examples above, each example is presented as the application of a fault, or test signal, to a single node. FIR decode block 4 can be programmed to apply multiple faults, based on the FIC applied to the FIR of FCR 2, through the application of a combination of signals FAULT_1 through FAULT_N to multiple node fault logic blocks, when signal ENABLE is asserted.

25 **[0021]** While the node fault logic block examples above illustrate stuck-at type faults, persons having skill in the art can easily program other fault types, such as stuck-open and bridging faults, into the integrated circuit.

30 **[0022]** While a preferred embodiment of the fault insertion system invention of Applicant has been particularly shown and described, it is not intended to be exhaustive nor limit the invention to the embodiment disclosed. It will be apparent to those of ordinary skill in the art that modifications to the preferred embodiment

